

Chapter 5: Enstore Commands

Enstore provides commands that allow you to communicate with various components of the system.

5.1 enstore file

This command communicates with the File Clerk (see section 8.1 *File Clerk*).

```
% enstore file [options]
```

Options:

- bfid <BFID> get info of a file
- h, --help prints this message (i.e., prints the options)
- list <VOLUME_NAME> lists the files in a volume
- ls-active <VOLUME_NAME> lists active files in a volume
- usage prints short help message

5.2 enstore library

This command communicates with the Library Manager (see section 8.4 *Library Manager*).

```
% enstore library [options] <library_name>
```

Options:

- get-asserts prints sorted lists of pending volume asserts
- get-queue <HOST_NAME> gets print queue submitted from the specified client (user) host; if empty string specified, print the whole queue
- get-suspect-vols prints suspect volume list
- get-work-sorted prints sorted lists of pending and active requests

-h, --help	prints this message (i.e., prints the options)
--usage	prints short help message

5.3 enstore monitor

This command communicates with the Monitor Server (see Chapter 9: *Monitoring Enstore on the Web*).

```
% enstore monitor [options]
```

User options:

-h, --help	prints this messge (i.e., prints the options)
--host <HOSTIP>	selects a single Enstore mover host
--usage	prints short help message
--verbose <VERBOSE>	prints out diagnostic information

5.4 enstore volume

This command communicates with the Volume Clerk (see section 8.2 *Volume Clerk*).

```
% enstore volume [OPTIONS]...
```

User options:

--gvol <VOLUME_NAME>	gets info of a volume in human readable time format
-h, --help	prints this messge (i.e., prints the options)
--just	used with --pvols to list problem
--list <VOLUME_NAME>	lists the files in a volume
--ls-active <VOLUME_NAME>	lists active files in a volume
--ls-sg-count	lists all sg counts
--pvols	lists all problem volumes
--usage	prints short help message
--vol <VOLUME_NAME>	gets info of a volume
--vols	lists all volumes

5.5 enstore pnfs

Enstore has a pnfs command that allows you to perform a variety of pnfs manipulations, as listed in the option table below. Off-site users cannot mount /pnfs, and therefore cannot run this command. The command syntax is:

```
% enstore pnfs [<option>]
```

The options are listed and defined in the following table. They all start with a double dash (--).

Note that the options that reset pnfs tag values should be used only by your experiment's designated Enstore guru(s).

List the options for the **enstore pnfs** command:

```
% enstore pnfs --help
```

Sample output:

```
USAGE: pnfs [ -h --bfid= --cat= --enstore-state= --file-family[=] --file-family-width[=] --file-family-wrapper[=] --files= --filesize= --help --info= --layer= --library[=] --pnfs-state= --tag= --tags --usage --xref= ]  
  
--bfid=<FILENAME>      lists the bit file id for file  
--cat=<FILENAME> [LAYER] see --layer  
--file-family=[FILE_FAMILY] gets file family tag, default; sets file  
family tag, optional  
--file-family-width=[FILE_FAMILY_WIDTH] gets file family width tag,  
default; sets file family tag,  
optional  
--file-family-wrapper=[FILE_FAMILY_WRAPPER] gets file family width tag,  
default; sets file family  
tag, optional  
--filesize=<FILE>      print out real filesize (Useful for files of size  
greater than (2G-1) bytes, since pnfs stores file  
size as 1 in this case.)  
-h, --help              prints this message  
--info=<FILENAME>      see --xref  
--layer=<FILENAME> [LAYER] lists the layer of the file  
--library=[LIBRARY]     gets library tag, default; sets library tag,  
optional  
--tag=<TAG> [DIRECTORY] lists the tag of the directory  
--tagchmod <PERMISSIONS> <TAG> changes the permissions for the tag;  
use UNIX chmod style permissions  
--tagchown <OWNER> <TAG> changes the ownership for the tag;  
OWNER can be "owner" or "owner.group".  
--tags=[DIRECTORY]      lists tag values and permissions  
--usage                prints short help message  
--xref=<FILENAME>      lists the cross reference data for file
```

List the tag values of specified directory (if no directory argument, it lists tags for cwd):

```
% enstore pnfs --tags [<directory>]
```

Sample output:

```
.(tag)(file_family) = dcache
.(tag)(file_family_width) = 1
.(tag)(file_family_wrapper) = cpio_odc
.(tag)(library) = eagle
.(tag)(storage_group) = test
-rw-rw-r-- 11 root sys 6 Jul 26 10:22 .(tag)(file_family)
-rw-rw-r-- 11 root sys 1 May 5 2000 .(tag)(file_family_width)
-rw-rw-r-- 11 root sys 8 May 5 2000 .(tag)(file_family_wrapper)
-rw-rw-r-- 11 root sys 5 May 5 2000 .(tag)(library)

-rw-r--r-- 11 root sys 4 Jul 26 10:20 .(tag)(storage_group)
```

List cross-reference information for specified file. The information includes:

- volume: storage media volume label
- location_cookie: file position on tape
- size: file size in bytes
- file_family: file family
- original_name: original name in /pnfs space; i.e., the destination filename given in the **encp** command used to copy the file to Enstore
- map_file: obsolete, but some older files may have a value here
- pnfsid_file: unique id for the file as assigned by PNFS
- pnfsid_map: obsolete, but some older files may have a value here
- bfid: unique id for the file as assigned by Enstore
- origdrive: id of drive used when file was written to media (files generated prior to 10/2000 will not have a value here)

```
% enstore pnfs --xref [/pnfs/../]<file>
```

Sample output:

```
volume: STORM2
location_cookie: 0000_00000000_0000001
size: 1024
file_family: zoo
original_name: /pnfs/mist/zaa/1KB_003
map_file:
pnfsid_file: 00010000000000000005E80
pnfsid_map:
bfid: WAMS100862864600000
origdrive: rain:/dev/rmt/tps0dln:0062025279
```

Set the value of the library tag (the virtual library associated with files in the directory) for the cwd:

```
% enstore pnfs --library=<value>
```

Set the value of the file family tag for the cwd (the file family does not have to be previously defined):

```
% enstore pnfs --file-family=<value>
```

Set the value of the file family width tag for the cwd (All directories having the same file family should also have the same width.):

```
% enstore pnfs --file-family-width=<value>
```

Set the value of the file family wrapper tag for the cwd. (All directories having the same file family must also have the same wrapper; changing the wrapper creates a new volume family which gets treated as a separate storage volume.):

```
% enstore pnfs --file-family-wrapper=<value>
```

